

Connect RZ/A3UL Evaluation Board Kit to your Azure IoT services

Table of Contents

- [Introduction](#)
- [Step 1: Prerequisites](#)
- [Step 2: Prepare the Device](#)
- [Step 3: Build SDK and Run Samples](#)
- [Step 4: Use Azure IoT Central](#)
- [Step 5: Integration with Azure IoT Explorer](#)
- [Step 6: Interact with IoT Plug and Play components](#)

Introduction

About this document

Azure Certified Device is no longer in service.

This document describes how to connect RZ/A3UL Evaluation Board Kit (EVK) to Azure IoT Central and Azure IoT Hub using the Azure IoT Explorer with certified device application and device models.

IoT Plug and Play certified device simplifies the process of building devices without custom device code. Using Solution builders can integrate quickly using the certified IoT Plug and Play enabled device based on Azure IoT Central as well as third-party solutions.

This getting started guide provides step by step instruction on getting the device provisioned to Azure IoT Central and Azure IoT Hub using Device Provisioning Service (DPS) and using Azure IoT Explorer to interact with device's capabilities.

Step 1: Prerequisites

You should have the following items ready before beginning the process:

- Prepare your development environment

Azure IoT SDK on RZ/A3UL EVK can be used on the following environment. Please check your environment before continuing.

Tools:

- IDE: e2 studio 2024-04 Windows 64-bit product version or later
- Tool Chain: GCC ARM A-Profile (Aarch64 bare-metal) 13.2.1.20231009
- Flexible Software Package (FSP): RZ/A FSP Packs V3.0.0

Refer to the following document to set up your development environment.

2. Starting Development Introduction - [Getting Started with RZ/A Flexible Software Package](#)
3. Set up an SMARC EVK - [Getting Started with RZ/A Flexible Software Package](#)

- Download the sample projects

Download the sample projects from the following URL.

[RZ/A3 Azure RTOS Example Programs](#)

- Start Serial terminal

Start a serial terminal program (such as PuTTY, HyperTerminal or Tera Term) using the following configuration:

- Baud Rate: 115200
- Data Bits: 8
- Parity: None
- Stop Bits: 1
- Flow Control: None
- COM Port: As shown in Windows™ Device Manager.

- Setup your IoT Central

Follow the steps below to setup your IoT Central.

1. Create the IoT Central application.
2. Register a new device in the IoT Central application.
3. Save the following values generated for newly registered device:

- **ID Scope**
- **Device ID**
- **Primary Key.**

Refer to the following document for the details.

[Quickstart - Connect a device to an Azure IoT Central application | Microsoft Learn](#)

Step 2: Prepare the Device

- Setup the device and connect power

1. Connect the CN14 connector of the board(RZ/A3UL EVK) and the PC with a USB cable.

Note: By connecting to CN14 via USB, it will be recognized by the PC as UART (CDC ACM).
Use the UART as a console.

2. Connect the CN8 connector of the board and an Ethernet port with the Ethernet cable.

connecting

- Add configuration
 1. Extract the sample project archive and import the projects into an empty e2studio workspace. Refer to the following document for the detail.
5.6 Importing an Existing Project into e2 studio - [Getting Started with RZ/A Flexible Software Package](#)
 2. Select the **RZA3UL_demo_azure_iot_pnp** sample and open **sample_config.h**.
 3. Set the Azure IoT device information constants to the values that you saved after you created Azure resources.

Constant name	Value
ENDPOINT	"global.azure-devices-provisioning.net"
ID_SCOPE	{Your ID Scope value}
REGISTRATION_ID	{Your Device ID value}
DEVICE_SYMMETRIC_KEY	{Your Primary Key value}



Step 3: Build SDK and Run Samples

Refer to following document to build and debug the project "RZA3UL_demo_azure_iot".

4.4 Build the Blinky Project - [Getting Started with RZ/A Flexible Software Package](#)

5.4 Debugging the Project - [Getting Started with RZ/A Flexible Software Package](#)

Note: Due to the bug of FSP, some project contents must be modified to execute this sample project. They have been already modified for the released project, but don't delete "rza" folder and regenerate it.

After running the projects, confirm that following message is displayed on the terminal software.


 terminal

Step 4: Use Azure IoT Central

IoT Central is an IoT application platform that reduces the cost and complexity of creating and managing IoT solutions.

- Verify the device status


To view the device status in IoT Central portal:

1. From the application dashboard, select **Devices** on the side navigation menu.
2. Check the **Device status** of the device is updated to **Provisioned**.
3. Check the **Device template** of the device has updated to **rza3ulevk**. provisioned

- View device telemetry

With IoT Central, you can view the flow of telemetry from your device to the cloud. In this sample project, Tj (temperature inside the LSI measured by thermal sensor unit (TSU)) is sent as telemetry.

To view telemetry in IoT Central portal:

1. From the application dashboard, select **Devices** on the side navigation menu.
2. Select the device from the device list.
3. View the telemetry as the device sends messages to the cloud in the **Overview** tab. overview

- Send commands

On the **Command** tab, you can send IoT Plug and Play command **getMaxMinReport**, **LED_ON** and **LED_OFF** to the device.

Note: If you check the function of command **LED_ON** or **LED_OFF**, connect Pmod(LED) to J1 connector(7:12) on the board.

centralcommand

- View raw data

On the **Raw data** tab, you can view the raw data such as **Telemetry** and **Command response**.

centralrawdata

Step 5: Integration with Azure IoT Explorer

Note: This section is advanced.

You can use the Azure IoT Explorer to view and manage the properties of your devices. In the following steps, you'll add a connection to your IoT hub in IoT Explorer. With the connection, you can view properties for devices associated with the IoT hub.

Download and install latest Azure IoT Explorer from the following URL.

<https://github.com/Azure/azure-iot-explorer/releases>

- Setup your IoT hub

Follow the steps below to setup your IoT hub.

1. Create an IoT hub.
2. Register a new device in the IoT hub.

Refer to the following document for the details.

[Use the Azure portal to create an IoT Hub | Microsoft Learn](#)

- Setup a DPS

The IoT Hub Device Provisioning Service (DPS) is a helper service for IoT Hub that enables zero-touch, just-in-time provisioning devices to the right IoT hub in a secure and scalable manner. In the following steps, you will enroll the board in DPS using Symmetric Key and provision it automatically in IoT Hub when connecting to the Internet.

Follow the steps below to setup a DPS.

1. Create a DPS to provision the device in IoT Hub automatically.
2. Link an IoT hub to the DPS instance.
3. Add an individual enrollment record in DPS. Your device can use it to connect to DPS and perform the provisioning in IoT Hub.
4. Save the following values:

- **ID Scope** of your Device Provisioning Service
- **Registration ID** for your device enrollment
- **Primary Symmetric Key** for your device enrollment.

Refer to the following documents for the details.

[Quickstart - Set up Device Provisioning Service in portal | Microsoft Learn](#)

[Manage device enrollments for Azure IoT Hub Device Provisioning Service in the Azure portal | Microsoft Learn](#)

- Add configuration
 1. Select the **RZA3UL_demo_azure_iot_pnp** sample in the e2studio workspace and open **sample_config.h**.
 2. Set the Azure IoT device information constants to the values that you saved after you created Azure resources.

Constant name	Value
ENDPOINT	"global.azure-devices-provisioning.net"
ID_SCOPE	{Your ID Scope value}
REGISTRATION_ID	{Your Registration ID value}
DEVICE_SYMMETRIC_KEY	{Your Primary Symmetric Key value}



- Build and run the sample

Follow [Step 3: Build SDK and Run Samples](#) to build and run the sample project.
- Setup the Azure IoT Explorer

To add a connection to your IoT hub:

1. Get the connection string for your IoT hub.
2. In Azure IoT Explorer, select **IoT hubs** > **Add connection**.
3. Paste the connection string into the **Connection string** box.
4. Select Save.



If the connection succeeds, the Azure IoT Explorer switches to a Devices view and lists your device.

To set IoT Plug and Play Settings:

5. In Azure IoT Explorer, select **IoT hubs** > **IoT Plug and Play Settings**.
6. Add **Public repository** to use IoT Plug and Play Settings.

 addrepository

To view device properties using Azure IoT Explorer:



1. Select the link for your device identity. IoT Explorer displays details for the devices.



2. Inspect the properties for your device in the **Device identity** panel.

Step 6: Interact with IoT Plug and Play components

Note: This section is advanced.

1. Select **IoT Plug and Play components** and **Default component**. pnpcomponents
2. Select **Properties(read-only)**. You can see the device properties. pnpproperties
3. Select **Commands**. You can send IoT Plug and Play command **getMaxMinReport**, **LED_ON** and **LED_OFF** to the device.

Note: If you check the function of command **LED_ON** or **LED_OFF**, connect Pmod(LED) to J1 connector(7:12) on the board.

pnpcommands